

Maintenir le système d'information en condition opérationnelle

Teddy MAHIETTE

HOUEDANOU Wilfried



Table des matières

Introduction.....	4
Glossaire et Terminologie.....	5
Présentation du Système d'Information.....	6
Contexte et Objectifs.....	6
Architecture	6
Mobile.....	6
Back-End.....	7
Rôles et Responsabilités	7
Équipe de développement.....	7
Équipe de Maintenance	8
Utilisateurs	8
Sécurité et Confidentialité.....	8
Maintenance du Système d'Information	9
Plan et Procédure de Maintenance	9
Objectifs de Maintenance	9
Procédures de Maintenance	9
Ressources Nécessaires.....	10
Système de Collecte des Incidents.....	10
Choix du Système de Monitoring.....	10
Bénéfices et Risques.....	11
Organisation de l'Équipe de Maintenance.....	11
Structure de l'Équipe.....	11
Coordination des Activités.....	11
Procédure de Gestion des Incidents	12
Identification et Classification des Incidents	12
Gestion des Incidents.....	12
Suivi et Amélioration.....	14
Impact Financier des Incidents	14
Estimation des Coûts.....	14
Estimation des Coûts pour des Cas Spécifiques.....	15
Formation des Utilisateurs	15
Documentation Utilisateur	15
Programmes de Formation	16
Gestion des Utilisateurs Finaux	16
Documentation Utilisateur	16
Formation des Utilisateurs	17
Gestion des Problèmes des Utilisateurs	18
Engagement des Utilisateurs.....	19
Évolution du Système d'Information.....	20

Organisation des Mises à jour Techniques.....	20
Identification des Acteurs et des Ressources.....	21
Propositions d'Évolution et Justifications.....	22
Conclusion	23

Introduction

La maintenance du système d'information est une composante essentielle pour assurer la continuité, la fiabilité et la performance d'une application en production. Dans le contexte de notre application nommée **Seeker**, dédiée à la prospection avec détecteurs de métaux, cette maintenance prend une importance particulière pour garantir une expérience utilisateur fluide et sans interruption.

Seeker combine des fonctionnalités avancées telles que la cartographie, l'enregistrement des découvertes et le partage, nécessitant une surveillance constante et une gestion proactive pour répondre aux défis opérationnels et techniques.

Ce document présente le plan de maintenance du SI pour cette application, avec pour objectif principal d'assurer que celle-ci fonctionne de manière optimale tout en répondant rapidement aux incidents et en intégrant les évolutions nécessaires. La maintenance du SI englobe plusieurs aspects clés, notamment l'organisation du travail de maintenance, la gestion des incidents, la formation des utilisateurs, et l'évolution technique du système.

La première partie de ce document décrit la structure et l'architecture, en mettant en lumière les rôles et les responsabilités des différents acteurs impliqués dans la gestion et la maintenance du SI.

Ensuite, nous détaillerons les procédures de maintenance, incluant les actions préventives et correctives, ainsi que les mécanismes de gestion des incidents.

La section suivante abordera l'impact financier potentiel des défaillances du système, en fournissant des estimations concrètes des coûts associés aux interruptions de service.

Nous examinerons également comment l'équipe de maintenance est organisée et coordonnée pour répondre efficacement aux incidents.

La formation des utilisateurs finaux est abordée en détail, avec des stratégies pour assurer qu'ils soient correctement formés et supportés.

Enfin, nous discuterons des processus de mise à jour et d'évolution du SI, afin de maintenir l'application à la pointe de la technologie et adaptée aux besoins futurs.

En fournissant une vue d'ensemble complète et structurée des pratiques et procédures de maintenance, ce document vise à garantir que Seeker reste performant, sécurisé et aligné avec les objectifs de ses utilisateurs.

Glossaire et Terminologie

Seeker	Chercheur
Prospection	Exploration d'un lieu pour trouver quelque chose
SI	Système d'information

Présentation du Système d'Information

Contexte et Objectifs

Seeker est une application mobile innovante conçue pour transformer l'expérience de la prospection avec détecteurs de métaux. L'application vise à offrir aux utilisateurs une plateforme intuitive pour découvrir, enregistrer et partager leurs trouvailles tout en facilitant la cartographie et la gestion des sites de prospection. En intégrant des fonctionnalités avancées telles que la visualisation des trouvailles sur une carte et l'accès à des informations détaillées sur les sites, Seeker s'adresse aux pratiquants de la détection de métaux, allant des amateurs aux experts.

Le système d'information se compose de deux principales technologies :

L'application mobile : Développée pour les plateformes iOS et Android, offrant une interface utilisateur conviviale et des fonctionnalités riches pour l'enregistrement des découvertes.

Le back-end : Basé sur Firebase, fournissant les services nécessaires pour le stockage, la gestion des utilisateurs, et le traitement des données.

Architecture

Mobile

L'application est développée à l'aide du Framework Flutter qui permet d'obtenir une application mobile disponible pour IOS et Android à partir d'une seule base de code.

Voici les principaux composants de l'application mobile :

- **Interface utilisateur** : Conçue pour offrir une expérience utilisateur fluide avec des écrans pour la cartographie, l'enregistrement des trouvailles et profils des utilisateurs.

- **Gestion des données locaux** : Utilisation de bases de données locales (par exemple, SQLite ou Room pour Android et Core Data pour iOS) pour stocker les données de manière temporaire et permettre l'accès hors ligne.

- **Intégration avec Firebase** : Les données sont synchronisées avec le back-end Firebase pour assurer la continuité et la disponibilité des informations en ligne.

Back-End

Le back-end de Seeker repose sur Firebase, une plateforme de développement d'applications proposée par Google. Firebase offre une suite de services pour gérer l'infrastructure et les opérations back-end, comprenant pour notre application :

Firebase Authentication : Gère l'inscription, la connexion et l'authentification des utilisateurs. Il prend en charge divers mécanismes de connexion, y compris les comptes par e-mail, les réseaux sociaux, et les identifiants anonymes.

Firebase Firestore : Une base de données NoSQL en temps réel utilisée pour stocker et synchroniser les données des utilisateurs, telles que les trouvailles, les cartes et les préférences. Firestore permet des requêtes rapides et une mise à jour en temps réel.

Firebase Storage : Permet le stockage et la gestion des fichiers, comme les images et les documents associés aux trouvailles. Ce service garantit que les fichiers sont stockés en toute sécurité et peuvent être récupérés rapidement.

Firebase Cloud Functions : Exécute des fonctions côté serveur en réponse à des événements dans l'application. Cela peut inclure des tâches comme l'envoi de notification push, le traitement des données en arrière-plan, et l'exécution de logiques métier.

Firebase Analytics : Collecte des données d'utilisation et fournit des rapports détaillés sur le comportement des utilisateurs, permettant ainsi d'optimiser l'application et de mieux comprendre les besoins des utilisateurs.

Rôles et Responsabilités

Équipe de développement

Développeurs Mobile : Responsables du développement de l'interface utilisateur sur les plateformes iOS et Android. Ils s'assurent que l'application est intuitive, réactive et conforme aux exigences fonctionnelles.

Développeurs Back-End : Responsables de la configuration et de la gestion des services Firebase, ainsi que de l'implémentation des API nécessaires pour la communication entre l'application mobile et le back-end.

Designers UI/UX : Conçoivent l'interface utilisateur et l'expérience utilisateur de l'application, veillant à ce que l'application soit visuellement attrayante et facile à utiliser.

Équipe de Maintenance

Administrateurs Firebase : Gèrent les configurations de sécurité, les règles d'accès, et surveillent la performance des services Firebase.

Techniciens Support : Fournissent un support aux utilisateurs finaux, résolvent les problèmes techniques et gèrent les incidents signalés.

Utilisateurs

Les passionnés de prospection utilisant Seeker pour enregistrer et visualiser partager leurs découvertes. Leurs retours sont cruciaux pour l'amélioration continue de l'application.

Sécurité et Confidentialité

L'application met un point d'honneur à assurer la sécurité des données des utilisateurs.

Les mesures de sécurité incluent :

- **Chiffrement des données** : Toutes les données sensibles sont chiffrées lors de leur transfert et stockage, garantissant leur confidentialité.
- **Contrôles d'accès** : Les règles de sécurité définies dans Firebase Authentication et Firestore assurent que seules les personnes autorisées peuvent accéder ou modifier les données.

Maintenance du Système d'Information

La maintenance du système d'information est essentielle pour garantir la disponibilité, la performance et la sécurité continues de l'application. Cette section décrit les procédures de maintenance, la gestion des incidents, les impacts financiers, l'organisation des équipes, et les stratégies de formation des utilisateurs.

Plan et Procédure de Maintenance

Objectifs de Maintenance

Le plan de maintenance de l'application vise à :

- **Assurer la Disponibilité** : Minimiser les interruptions de service et garantir la continuité de l'application.
- **Améliorer la Performance** : Optimiser les performances de l'application pour offrir une expérience utilisateur fluide.
- **Garantir la Sécurité** : Protéger les données des utilisateurs contre les menaces et les vulnérabilités.
- **Faciliter l'évolution** : Intégrer les nouvelles fonctionnalités et les améliorations basées sur les retours des utilisateurs.

Procédures de Maintenance

Notre maintenance préventive se basera sur deux points, des mises à jour régulières ainsi que des révisions périodiques. Les mises à jour de l'application pour les plateformes IOS, Android pour la partie mobile et Firebase pour la partie back-end permettront d'obtenir des améliorations d'un point de vue sécurité, mais également toutes les nouvelles fonctionnalités à venir.

Pour notre maintenance corrective, il y a tout d'abord une surveillance continue de l'application grâce aux outils Firebase, plus précisément Crashlytics et Performance qui nous permettent de détecter les anomalies, les dysfonctionnements, etc. Une fois les bugs et incidents identifiés, la résolution a lieu après une phase de concertation avec l'équipe afin de déterminer les causes et les solutions.

Pour la dernière procédure qui est la maintenance évolutive, nous nous baserons tout d'abord sur les feedbacks des utilisateurs afin de déterminer les améliorations ainsi que les nouvelles fonctionnalités à mettre en place. Le retour des utilisateurs sera également important afin de valider les nouvelles fonctionnalités avant leurs déploiements officiels, cela permettra de garantir le bon fonctionnement et une intégration harmonieuse avec les fonctionnalités existantes. Pour ce faire, nous pouvons imaginer une version bêta pour les utilisateurs qui le souhaitent.

Ressources Nécessaires

Pour assurer une maintenance efficace, les ressources suivantes sont nécessaires :

Équipe Technique : Développeurs back-end, développeurs front-end, ingénieurs en sécurité, et designers UX/UI.

Outils de Surveillance : Solutions de monitoring pour les performances de l'application et les services Firebase, telles que Firebase Crashlytics et Google Analytics.

Documentation : Guides et procédures documentées pour les processus de maintenance et de résolution des incidents.

Système de Collecte des Incidents

Pour assurer une surveillance efficace et la gestion des incidents dans Seeker, qui utilise principalement Firebase pour son back-end, il est crucial de choisir un système de collecte des incidents adapté. Voici le détail de la mise en place et des outils utilisés pour la surveillance et la gestion des incidents.

Choix du Système de Monitoring

Pour surveiller les performances et la disponibilité de notre application mobile basée sur Firebase, nous avons opté pour les outils suivants :

- **Firebase Crashlytics** : Permet de suivre les crashes de l'application en temps réel, d'analyser les problèmes de stabilité et de comprendre les erreurs rencontrées par les utilisateurs. L'intégration native avec Firebase, offre une vue unifiée des erreurs et des problèmes de performance directement dans le même environnement que le backend. Il fournit des rapports détaillés sur les crashes, ce qui facilite le diagnostic et la correction rapide des problèmes.
- **Firebase Performance** : Offre une visibilité sur les performances de l'application en mesurant des indicateurs tels que le temps de réponse des requêtes réseau et le temps de démarrage de l'application. Cet outil intégré avec Firebase, il fournit des métriques précieuses sur les performances de l'application et aide à identifier les goulots d'étranglement dans les requêtes réseau ou dans les opérations critiques.

Le dernier outil utilisé est **Google Cloud Monitoring** (anciennement Stackdriver). Celui-ci offre une surveillance approfondie des ressources et des services cloud y compris les services Firebase. Cela nous permettra d'avoir une vision de l'infrastructure cloud sous-jacente et d'intégrer les métriques de performance et les journaux des services Google Cloud avec Firebase. Ces outils offrent des fonctionnalités avancées pour détecter et signaler rapidement les anomalies, assurant ainsi une réponse proactive aux problèmes.

Bénéfices et Risques

Bénéfices :

- Détection proactive des incidents : Les outils permettent une surveillance en temps réel, ce qui aide à détecter les problèmes avant qu'ils n'affectent gravement les utilisateurs.
- Réductions des temps d'arrêt : La capacité à identifier et résoudre rapidement les problèmes contribue à minimiser les interruptions de service et les temps d'arrêt de l'application.
- Amélioration de la réactivité : Les alertes et les rapports détaillés permettent aux équipes techniques de réagir rapidement aux incidents, ce qui améliore la stabilité générale de l'application.

Risques :

- Dépendance à l'outil choisi : Une forte dépendance à un outil spécifique peut entraîner des problèmes en cas de défaillance ou de limitation de cet outil. Il est important de bien évaluer et de tester les outils choisis pour garantir leur adéquation avec les besoins.
- Possibilité de faux positifs/négatifs : Les outils de monitoring peuvent parfois générer des faux positifs (alerte de problème inexistant) ou des faux négatifs (problèmes non détectés), ce qui peut influencer la qualité de la gestion des incidents.

Organisation de l'Équipe de Maintenance

Structure de l'Équipe

L'équipe de maintenance est composée d'une équipe technique et de techniciens support.

L'équipe technique composée de développeurs back-end pour gérer les services Firebase, de développeurs front-end pour résoudre les problèmes d'interface et d'ingénieurs en sécurité pour assurer la protection des données. Chaque membre de l'équipe est assigné à des tâches spécifiques en fonction de leurs expertises pour assurer une gestion efficace des incidents et des tâches de maintenance.

Les techniciens support auront pour but de fournir une assistance aux utilisateurs finaux afin de gérer leurs demandes et résoudre les problèmes signalés.

Coordination des Activités

Afin d'assurer une coordination efficace entre les membres de l'équipe, des réunions hebdomadaires afin de faire un point sur les problèmes et les tâches de maintenance en cours. À l'heure actuelle, ces réunions sont réalisées sur discord pour faciliter la collaboration et le partage d'informations.

Procédure de Gestion des Incidents

La gestion des incidents est cruciale pour garantir la continuité de service et la satisfaction des utilisateurs de l'application Seeker. Cette procédure détaillée couvre l'ensemble du cycle de vie d'un incident, depuis son identification jusqu'à son suivi post-résolution, en passant par les étapes de classification, de gestion et d'amélioration continue.

Identification et Classification des Incidents

L'application utilise Firebase pour surveiller en temps réel ses performances et identifier les incidents. Ces incidents peuvent inclure des crashes d'application, des ralentissements, des erreurs d'interface utilisateur, ou des problèmes de connectivité. Par exemple, si un utilisateur signale que l'application plante à n'importe quel moment Firebase Crashlytics captera ce crash, générera un rapport, et notifiera l'équipe technique.

Une fois un incident identifié, il est classé en fonction de sa gravité. Voici comment ces catégories sont définies :

- Critique : Incidents qui empêchent totalement l'utilisation de l'application. Par exemple, si tous les utilisateurs sont incapables de se connecter à l'application ou si une mise à jour introduit un bug qui cause un crash généralisé.
- Élevée : Problèmes affectant une fonctionnalité majeure, mais ne rendant pas l'application inutilisable. Par exemple, l'impossibilité d'enregistrer une découverte pour une certaine région géographique.
- Moyenne : Problèmes affectant des fonctionnalités secondaires ou causant des inconvénients mineurs. Par exemple, un dysfonctionnement dans la personnalisation du profil utilisateur.
- Basse : Incidents mineurs sans impact significatif sur l'expérience utilisateur globale. Par exemple, une faute d'orthographe dans les messages d'aide de l'application.

Ces classifications sont cruciales pour prioriser les efforts de résolution et communiquer l'importance de chaque incident aux parties prenantes.

Gestion des Incidents

Lorsqu'un incident est détecté, une notification est envoyée dans le canal Discord dédié, nommé **#incident**. Par exemple, si une panne critique est détectée affectant la connexion des utilisateurs, un message type sera posté détaillant l'incident.

Exemple d'un message d'incident :

Gravité : Critique

Description : Impossible pour les utilisateurs de se connecter à l'application.

Impact : Affecte 100% des utilisateurs.

Cause présumée : Problème de serveur après la mise à jour X.X.X.

Action immédiate : Investigation en cours par l'équipe backend.

Prochaine mise à jour : Dans 30 minutes.

Les membres de l'équipe concernés seront immédiatement alertés via des notifications push et des mentions directes pour garantir une réponse rapide.

Étapes de Gestion : La gestion de l'incident suit trois étapes clés :

1. **Diagnostic :** L'équipe technique analyse les logs fournis par Firebase et les retours des utilisateurs pour identifier la cause profonde de l'incident. Par exemple, dans le cas d'une panne de connexion, l'équipe pourrait découvrir que le problème provient d'une surcharge du serveur après une campagne de marketing.
2. **Correction :** Une fois la cause identifiée, une solution est développée. Cela pourrait impliquer de déployer un correctif, d'ajuster la configuration du serveur, ou de revenir à une version précédente du logiciel. Par exemple, si un bug dans la mise à jour 2.1.0 est identifié comme la cause, l'équipe pourrait rapidement développer et déployer une version 2.1.1 pour résoudre le problème.
3. **Validation :** Après la correction, l'équipe effectue des tests approfondis pour s'assurer que l'incident est bien résolu et que l'application fonctionne normalement sans effets secondaires indésirables. Par exemple, une fois le correctif appliqué, des tests de charge sont effectués pour vérifier que la solution tient sous pression.

Pendant toute la durée de la gestion de l'incident, l'équipe communique régulièrement avec les utilisateurs via des notifications dans l'application et des messages sur les réseaux sociaux. Par exemple, un message pourrait être envoyé :

« Nous avons identifié un problème affectant la connexion à notre application. Nos équipes travaillent activement afin de résoudre ce problème. Nous vous tiendrons informés de l'avancement. Merci pour votre patience. »

Une fois l'incident résolu et validé, il est formellement clôturé. Un rapport de clôture est publié dans le canal **#incident** de Discord, incluant les détails de l'incident, les actions entreprises, et le temps total de résolution. La documentation technique et les procédures internes sont mises à jour pour refléter les leçons apprises.

Suivi et Amélioration

Après chaque incident, une réunion post-mortem est organisée pour analyser ce qui s'est passé, ce qui a bien fonctionné, et ce qui peut être amélioré. Un rapport détaillé est rédigé, comprenant :

- **Causes identifiées** : Par exemple, une surcharge du serveur à la suite d'une campagne marketing non anticipée.
- **Solutions apportées** : Par exemple, l'ajout de capacités de serveur et l'amélioration de la surveillance en temps réel.
- **Recommandations** : Par exemple, planifier des tests de charge avant toute grande campagne marketing et améliorer les outils de notification pour alerter les utilisateurs plus rapidement.

Les équipes sont également formées en continu sur les meilleures pratiques de gestion des incidents, et les procédures sont révisées régulièrement pour intégrer les leçons apprises. Par exemple, après plusieurs incidents critiques causés par des mises à jour, il est décidé de mettre en place une phase de test bêta plus longue pour les nouvelles fonctionnalités.

Ainsi, chaque incident devient une opportunité d'amélioration, renforçant la robustesse du système et la satisfaction des utilisateurs.

Impact Financier des Incidents

Estimation des Coûts

L'impact financier d'une défaillance dans notre application Seeker peut être significatif, en particulier pour des fonctionnalités critiques telles que la cartographie des trouvailles.

Les coûts associés peuvent inclure :

Perte de Revenus : Si l'application rencontre un problème majeur, il peut y avoir une perte de revenus due à une baisse des téléchargements ou à des achats in-app manqués.

Coûts de Réparation : Les coûts liés à la correction des bugs, y compris les heures de développement et les ressources nécessaires pour résoudre les problèmes.

Pertes de Réputation : Les problèmes techniques peuvent affecter la perception de l'application par les utilisateurs et entraîner des avis négatifs, ce qui peut impacter négativement la croissance et la rétention des utilisateurs.

Estimation des Coûts pour des Cas Spécifiques

Fonctionnalité de Cartographie

Problème : Si la fonctionnalité de cartographie ne fonctionne pas, les utilisateurs ne pourront pas visualiser leurs trouvailles sur la carte, ce qui pourrait entraîner une perte d'engagement et de satisfaction.

Coût estimé : La perte de revenus peut être estimée en fonction du nombre d'utilisateurs actifs et du taux de conversion pour les achats in-app liés à la cartographie.

Problèmes de Performance Générale

Problème : Des lenteurs ou des interruptions de service peuvent réduire l'accessibilité de l'application.

Coût estimé : Inclut les coûts de réparation, ainsi que la perte de revenus potentielle due à l'interruption du service.

Formation des Utilisateurs

Une formation des utilisateurs finaux est importante afin qu'ils puissent aisément utiliser l'application et qu'ils ne rencontrent pas de blocage, cela évite également certaines sollicitations des équipes support sur des points qu'une formation pourrait éviter.

Documentation Utilisateur

Une documentation pour les utilisateurs a pour but de fournir un guide clair et concis pour les aider à comprendre et à utiliser l'application efficacement. Cette documentation sera disponible en ligne via un site vitrine, celle-ci contiendra également un onglet FAQ.

L'équipe support aura aussi pour mission de répondre aux questions des utilisateurs et résoudre leurs problèmes.

Programmes de Formation

Formation Initiale

Public ciblé : Nouveaux utilisateurs de l'application.

Planification : Sessions de formation en ligne ou en personne, animées par des experts de l'application.

Formation Continue

Public cible : Utilisateurs existants, pour les nouvelles fonctionnalités ou les mises à jour majeures.

Planification : Webinaires réguliers et mises à jour de la documentation en fonction des nouvelles versions de l'application.

Gestion des Utilisateurs Finaux

La gestion des utilisateurs finaux est cruciale pour assurer une adoption réussie de Seeker, une application mobile dédiée à la prospection avec détecteurs de métaux. Cela inclut non seulement l'introduction de l'application aux utilisateurs, mais aussi la fourniture d'un soutien continu pour optimiser leur expérience. Voici un aperçu détaillé des stratégies et des procédures mises en place pour gérer efficacement les utilisateurs finaux de l'application.

Documentation Utilisateur

La documentation utilisateur vise à fournir aux utilisateurs toutes les informations nécessaires pour utiliser efficacement l'application. Elle comprend des guides détaillés, des FAQ, et des tutoriels pour répondre aux questions courantes et faciliter la prise en main de l'application.

Eventuelle composition de la documentation :

1 - Guide de Démarrage Rapide :

Contenu : Instructions étape par étape pour installer l'application, créer un compte, et commencer à utiliser les fonctionnalités principales.

Format : Document PDF téléchargeable et pages dédiées sur le site web de l'application.

2 - Manuel de l'Utilisateur :

Contenu : Description détaillée de toutes les fonctionnalités de l'application, y compris la cartographie, la gestion des trouvailles, et les paramètres de profil.

Format : Guide interactif disponible dans l'application et en ligne.

3 - FAQ (Questions Fréquemment Posées) :

Contenu : Réponses aux questions courantes concernant les fonctionnalités, les problèmes techniques, et les questions de sécurité.

Format : Section dédiée sur le site web et dans l'application.

4 - Tutoriels Vidéo :

Contenu : Vidéos explicatives sur les fonctionnalités de l'application, la configuration initiale, et les meilleures pratiques.

Format : Hébergé sur la plateforme YouTube et intégré dans l'application.

Formation des Utilisateurs

Les formations sont conçues pour aider les utilisateurs à se familiariser rapidement avec l'application, maximiser leur engagement et résoudre les problèmes potentiels dès le départ.

Eventuelle plan de formation :

1 - Sessions de Formation Initiales :

Public Cible : Nouveaux utilisateurs, partenaires et équipes internes.

Dates : Organisées lors du lancement de l'application et des mises à jour majeures.

Format : Ateliers en ligne (Webinars) et sessions en présentiel, selon la disponibilité des participants.

2 - Formation Continue :

Public Cible : Utilisateurs existants cherchant à approfondir leurs connaissances ou à découvrir de nouvelles fonctionnalités.

Dates : Organisée trimestriellement ou lors de la publication de nouvelles fonctionnalités.

Format : Ateliers interactifs, tutoriels avancés, et sessions de questions-réponses en ligne.

3 - Supports de Formation :

Documents : Guides et manuels distribués lors des sessions de formation.

Vidéo : Enregistrements des sessions disponibles en ligne pour une consultation ultérieure.

Assistance : Support en ligne via un chat, email, et forums pour répondre aux questions post-formation.

4 - Formation des Administrateurs :

Public Cible : Administrateurs et gestionnaires d'équipe qui ont besoin de gérer plusieurs comptes ou d'exploiter des fonctionnalités avancées.

Dates : Pré-lancement et mises à jour importantes.

Format : Séances individuelles ou en groupe pour une formation personnalisée.

Gestion des Problèmes des Utilisateurs

Pour cette partie traitant de la gestion des problèmes des utilisateurs, nous avons mis en place une gestion. Le but est de fournir un soutien rapide et efficace afin de résoudre les problèmes rencontrés par les utilisateurs et garantir une expérience sans interruption.

Notre gestion se découpe en trois points :

- Canaux de support : Les utilisateurs pourront nous faire part de leurs différents problèmes via tout d'abord une adresse Email dédiée, un forum de support et par la suite lors de l'agrandissement du système d'information d'un chat en direct.
- Processus de résolution : Tout d'abord, les utilisateurs soumettent leurs problèmes via les différentes manières cités ci-dessus ensuite, la première étape dans notre équipe sera un tri des demandes en fonction de la gravité et du type du problème. Viens après la résolution du problème qui se suit d'un retour utilisateur permettant d'obtenir un avis sur les points d'amélioration et identifier les tendances des problèmes récurrents.
- Suivie et Amélioration : Il sera réalisé régulièrement des revues afin d'identifier les causes sous-jacentes et les tendances des différents incidents rencontrés. Cela sera suivi d'une mise à jour de la documentation et des formations afin d'éviter si possible une récurrence des problèmes rencontrés par les utilisateurs.

Engagement des Utilisateurs

Maintenir un engagement élevé des utilisateurs est notre objectif. Pour ce faire, nous leur offrons des moyens de se connecter avec la communauté et d'améliorer continuellement leur expérience de l'activité de prospection via notre application Seeker.

Nous avons imaginé différents moyens d'accroître cet engagement :

- Programmes de fidélité et récompenses : Offrir des récompenses aux utilisateurs qui atteignent certains jalons, tels que la découverte d'un certain nombre de trouvailles ou la participation à des défis communautaires. Cela permettra d'encourager les utilisateurs à explorer toutes les fonctionnalités de l'application et à rester actifs dans leurs activités et dans l'utilisation de Seeker.
- Événements communautaires : Évènements en ligne et/ou en personnes, tels que des concours de détection ou des rencontres de passionnés de détection, afin de créer des opportunités pour les utilisateurs de se rencontrer, de partager leurs découvertes et d'échanger des conseils.
- Sondages et feedback : Sondages en ligne, questionnaires intégrés dans l'application et sessions de retour d'expérience. Afin d'obtenir les avis des utilisateurs leurs souhaits sur l'ajout de nouvelles fonctionnalités intéressantes afin de leur procurer le sentiment d'appartenance et d'intégration à l'application.

Évolution du Système d'Information

L'évolution continue du Système d'Information est essentielle pour maintenir la pertinence, l'efficacité et la compétitivité de Seeker. Cette partie présente les processus et les stratégies pour piloter l'évolution du SI, y compris la planification des mises à jour, l'identification des besoins futurs, et la gestion des ressources nécessaires pour les évolutions.

Organisation des Mises à jour Techniques

Objectif: Assurer une mise à jour régulière et coordonnée des fonctionnalités techniques de l'application pour répondre aux besoins des utilisateurs et intégrer les dernières innovations technologiques.

Étape de planification des mises à jour :

1 - Identification des Besoins

Sources d'Information: Feedback des utilisateurs, rapports de performance, tendances du marché, et évolutions technologiques.

Évaluation des Besoins : Analyse des demandes des utilisateurs et des lacunes identifiées dans les versions actuelles de l'application.

2 - Définition des Objectifs

Objectifs Techniques : Amélioration des performances, ajout de nouvelles fonctionnalités, et résolution des problèmes existants.

Objectifs Utilisateurs : Répondre aux attentes des utilisateurs, améliorer l'expérience utilisateur, et intégrer des suggestions reçues.

3 - Planification des Versions

Calendrier des Mises à Jour : Établissement d'un calendrier pour les mises à jour majeures et mineures. Les mises à jour majeures peuvent inclure des fonctionnalités nouvelles ou améliorées, tandis que les mises à jour mineures se concentreront sur des corrections de bugs et des améliorations de performance.

Jalons de Développement : Définition des phases de développement, des tests, et de la mise en production. Chaque mise à jour devra être validée par des tests rigoureux avant son déploiement.

4 - Communication des Changements

À l'Équipe : Réunions régulières avec les équipes de développement, de support et de marketing pour coordonner les efforts et s'assurer que tout le monde est informé des changements à venir.

Aux Utilisateurs : Annonces via les canaux de communication de l'application, tels que les notifications dans l'application, les newsletters, et les mises à jour sur le site web.

Étape des processus de Mise à Jour :

1 - Développement :

Conception : Développement de nouvelles fonctionnalités en fonction des objectifs définis.

Tests : Réalisation de tests fonctionnels, de performance et de sécurité pour garantir la stabilité et la qualité des nouvelles fonctionnalités.

2 - Déploiement :

Phase Pilote : Déploiement initial auprès d'un groupe restreint d'utilisateurs pour recueillir des retours avant le lancement complet.

Lancement Complet : Mise en production des nouvelles fonctionnalités pour tous les utilisateurs.

3 - Suivi Post-Déploiement :

Surveillance : Surveillance des performances et de l'utilisation des nouvelles fonctionnalités.

Support : Fourniture de support pour résoudre tout problème rencontré par les utilisateurs à la suite de la mise à jour.

Identification des Acteurs et des Ressources

Objectif : Garantir que les ressources nécessaires sont disponibles pour le développement, le déploiement et le support des mises à jour.

Acteurs Clés :

- Équipe de Développement :

Rôles : Développeurs, ingénieurs en logiciel, architectes techniques.

Responsabilités : Conception et développement des nouvelles fonctionnalités, intégration des changements, et résolution des problèmes techniques.

- Équipe de Test :

Rôles : Testeurs QA, ingénieurs en tests.

Responsabilités : Réalisation des tests fonctionnels, de performance et de sécurité.

- Équipe de Support :

Rôles : Support technique, gestionnaires de la relation utilisateur.

Responsabilités : Assistance aux utilisateurs, gestion des incidents post-déploiement.

- **Équipe de Gestion de Projet :**

Rôles : Chefs de projet, coordinateurs de projet.

Responsabilités : Planification, gestion des jalons et coordination des ressources.

Ressources nécessaires :

- **Outils de Développement :**

Environnements de développement : IDE, plateformes de développement mobile.

Outils de collaboration : Plateformes de gestion de projets, outils de communication.

- **Outils de Test :**

Outils de Tests Automatisés : Frameworks et outils pour les tests automatisés.

Environnements de Test : Simulateurs, émulateurs et appareils physiques pour les tests.

- **Ressources Humaines :**

Budget : Allocation budgétaire pour le développement, le test et le support.

Formation : Formation continue pour l'équipe sur les nouvelles technologies et méthodologies.

Propositions d'Évolution et Justifications

Une évolution du SI est importante pour suivre les tendances du marché et continuer à satisfaire les utilisateurs selon leurs besoins. C'est pourquoi nous avons en tête certaines évolutions.

Propositions d'Évolution :

Intégration de nouvelles technologies

L'intégration de nouvelles technologies permettra l'amélioration de l'expérience utilisateur en rendant la prospection plus interactive et informative. On peut imaginer, si possible, l'utilisation de la réalité augmentée ou autres technologies pour répondre à ce besoin.

Optimisation des Performances

Il est important que l'application reste utilisable facilement et sans frustration pour les utilisateurs provoquée par des ralentissements ou des temps d'attente trop longs. En réduisant ces facteurs, nous augmenterons la satisfaction et le confort de nos utilisateurs.

Extension des Fonctionnalités

Un des objectifs majeurs de notre application est de créer une communauté autour de l'activité de prospection, c'est pour cela que nous avons comme objectif par la suite de créer un réseau social dans Seeker afin de faciliter la communication, le partage des découvertes par exemple et la propagation de cette activité entre nos utilisateurs actuels et futurs.

Conclusion

En conclusion, un plan de maintenance solide est crucial pour le bon fonctionnement et la pérennité de notre application Seeker. La mise en œuvre d'une maintenance proactive, associée à des procédures rigoureuses de gestion des incidents, permet non seulement de minimiser les interruptions de service, mais aussi de maintenir un niveau de satisfaction élevé parmi les utilisateurs. La capacité à répondre rapidement aux incidents, à former efficacement les utilisateurs et à gérer les évolutions techniques est essentielle pour garantir la résilience et la compétitivité de l'application.

Le document a détaillé les diverses facettes de la maintenance du SI, allant de l'organisation du travail de maintenance à la gestion des incidents, en passant par l'analyse de l'impact financier et la coordination des équipes. En mettant en place des procédures bien définies et en assurant une communication claire, le système d'information autour de Seeker peut anticiper et gérer les défis techniques tout en optimisant l'expérience utilisateur.

Les recommandations formulées dans ce plan visent à renforcer la stabilité du système et à préparer l'application pour l'avenir en intégrant des mises à jour et des améliorations continues. La gestion efficace de la maintenance et des évolutions garantit non seulement la performance continue de l'application, mais aussi son alignement avec les attentes des utilisateurs et les standards technologiques actuels.

Ainsi, en adhérant à ce plan de maintenance, le SI s'assure de rester une solution fiable et innovante dans le domaine de la prospection, répondant aux besoins évolutifs de ses utilisateurs tout en optimisant ses ressources et ses capacités techniques.